

BAB V

IMPLEMENTASI SISTEM PAKAR

Pada bab ini, program aplikasi diimplementasikan untuk dapat dioperasikan pada kondisi yang sebenarnya, sehingga akan dapat dipastikan apakah program aplikasi ini dapat mencapai tujuan yang diinginkan. Sebelum masuk ke dalam implementasi perangkat lunak, terlebih dahulu akan dipaparkan spesifikasi sistem perangkat lunak (*software*) maupun perangkat keras (*hardware*) yang dibutuhkan untuk mendukung program aplikasi tersebut.

5.1 Implementasi Kebutuhan Sistem

Pada bagian spesifikasi kebutuhan sistem ini, diulas tentang kebutuhan sistem perangkat lunak maupun perangkat keras yang dapat mendukung dalam pembuatan maupun pada saat pengoperasian program aplikasi.

5.1.1 Implementasi Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan agar program aplikasi tersebut dapat berjalan dengan baik, minimal harus memenuhi spesifikasi sebagai berikut :

1. *Intel Pentium 90* atau yang lebih tinggi.
2. *Microsoft Windows 95* atau yang lebih tinggi.
3. *Hard disk 5409 Mbyte* atau lebih besar.
4. *RAM 32 M* atau lebih besar.
5. *Monitor VGA* atau SVGA.
6. Kartu VGA minimal 1 Mbyte
7. *Mouser*

8. *Keybord*

5.1.2 Implementasi Perangkat Lunak (*Software*)

Perangkat lunak yang dibutuhkan untuk mendukung dalam pembuatan atau pengoperasian program aplikasi ini, sebagai berikut :

1. Sistem operasi *Windows*.

Karena bahasa pemrograman yang digunakan untuk membuat program adalah berbasis *windows*, dan selain itu *windows* memiliki antarmuka grafis (*Graphical User Interface / GUI*) yang menawarkan kerja unik dan *user friendly*, bila dibandingkan dengan lingkungan DOS yang bekerja dengan baris-baris perintah

2. *Borland Delphi*

Untuk membuat program digunakan bahasa pemrograman *Borland Delphi 5.0*. Karena *Delphi* menyediakan fasilitas-fasilitas yang luas mulai dari fungsi untuk membuat *form* hingga fungsi untuk menggunakan beberapa *format file* basis data yang populer (*dbase, paradox, dan sebagainya*). *Delphi* mempermudah pembuatan program bagi komponen-komponen *windows* seperti *label, button* dan bahkan *dialog*, sehingga mempermudah dalam pembuatan antarmuka yang bersahabat dengan pengguna.

5.2 Batasan Implementasi

Perangkat lunak yang dibuat memiliki batasan-batasan dalam implementasinya yang dapat lebih mengarahkan bentuk program aplikasi sesuai dengan rancangan program. Dalam pengembangan pembuatan perangkat lunak ini, diasumsikan bahwa

perangkat lunak yang dibuat merupakan suatu program aplikasi (*software*) yang dapat digunakan untuk melakukan mengetahui jenis kepribadian seseorang sesuai dengan batasan-batasan seperti yang dijelaskan pada batasan masalah. Dimana jenis kepribadian yang telah didiagnosa, akan ditampilkan namanya disertai dengan keterangan, penyebab dan solusi dalam menghadapi jenis kepribadian tersebut.

Program ini dibangun dengan menggunakan tampilan yang diharapkan dapat mempermudah pengguna dalam pengoperasian program. Dengan asumsi bahwa setiap pengguna yang mengoperasikan program tidak harus orang yang terbiasa menggunakan komputer.

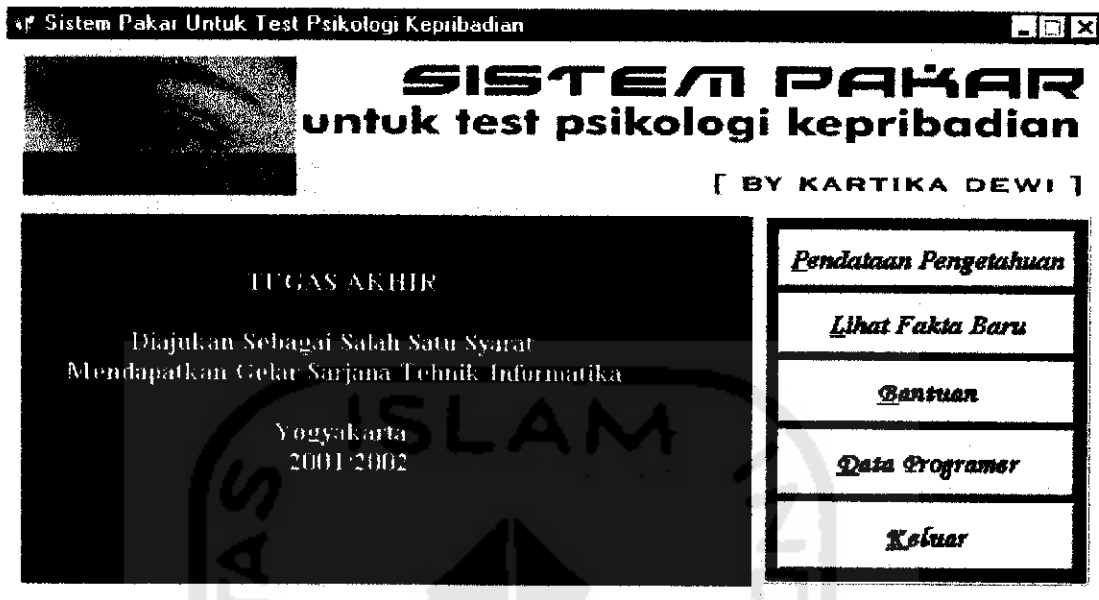
5.3 Implementasi

Implementasi test psikologi dengan menggunakan sistem pakar dibagi menjadi tiga yaitu implementasi untuk konsultasi, implementasi untuk pakar dan implementasi untuk hasil keluaran sistem(*output*)

5.3.1 Implementasi untuk Pakar

5.3.1.1 Implementasi Form Tampilan Awal untuk Pakar

Fungsi dari tampilan awal adalah untuk menghubungkan ke form aplikasi sistem diagnosa kepribadian manusia. Form menu utama terdiri dari empat buah tombol yang berlainan fungsinya. Tampilan utama dari sistem pakar untuk mendiagnosa jenis kepribadian manusia terlihat pada gambar 5.1



Gambar 5.1 Form tampilan awal untuk *pakar*

Langkah-langkah dalam pembuatan *form tampilan awal* :

1. Dari menu *file*, pilih menu *new application*
2. Membuat nama form
 - a. klik *object inspector, propertis, option*
 - b. ketikan nama form sesuai dengan kebutuhan

Membuat image



Image berfungsi untuk menempatkan gambar ke dalam sebuah aplikasi program yang dibuat.

1. klik *komponen pallette additional*
2. klik *image, form*
3. Pada *object inspector, propertis, picture* diisi gambar yang akan ditampilkan
4. Atur letak dan ukuran gambar sesuai dengan kebutuhan dan keinginan

Membuat tombol 'PENDATAAN PENGETAHUAN'

Pendataan Pengetahuan

Fungsinya untuk masuk ke dalam form selanjutnya yang berisikan tentang pengisian data yang dilakukan oleh seorang pakar.

1. klik *komponen pallette additional, speedbutton*, dan pindahkan ke form
2. klik *object inspector, caption* isikan tulisan PENDATAAN PENGETAHUAN
3. Pada *events onklik* isikan program berikut

```
procedure Tf_muka.SpeedButton1Click(Sender: TObject);
begin
    speedbutton1.enabled:=false;
    speedbutton2.enabled:=false;
    speedbutton3.enabled:=false;
    speedbutton4.enabled:=false;
    f_input.ShowModal;
    {f_soal.ShowModal;}
end;
```

Membuat tombol 'BANTUAN'

Bantuan

Fungsinya untuk melihat bantuan yang telah dibuat agar bagi pengguna yang belum tahu tentang cara penggunaan sistem ini. Dengan adanya bantuan ini diharapkan mampu membantu seseorang yang tidak mengerti komputer dalam menjalankan program ini.

Cara pembuatannya sama dengan pembuatan tombol diatas.

Membuat tombol 'DATA PROGRAMER'

Data Programmer

Fungsinya untuk melihat data dari pembuat program. Cara pembuatannya juga sama cuman dalam *event onklik* masukan listing program berikut

```
procedure Tf_muka.SpeedButton2Click(Sender: TObject);
begin
    aboutbox.showmodal;
end;
```

Membuat tombol 'KELUAR'

Keluar

Fungsinya untuk keluar dari program ini. Cara pembuatannya sama tetapi dalam *event onklik* masukan listing program berikut :

```
procedure Tf_muka.SpeedButton4Click(Sender: TObject);
begin
  f_muka.Close;
end;
```

5.3.1.2 Implementasi Form Untuk Pakar Menginputkan Data Nama Dan Ciri Jenis Kepribadian

Proses ini digunakan untuk pengisian *data base* dari seorang pakar. Disini seorang pakar dituntut untuk *objektif* terhadap apa yang akan diisikan ke dalam *data base*. Karena apabila terjadi kesalahan di dalam pengisian *data base* maka akan mempengaruhi hasil akhir yang akan ditampilkan oleh sistem. Pada tampilan ini berisi nama dan ciri-ciri jenis kepribadian. Banyak atau tidaknya data base yang dimasukan ke dalam data base tergantung dari banyak atau tidaknya fakta yang ditemukan. Tampilan identifikasi jenis kepribadian dapat dilihat pada gambar 5.2.

Gambar 5.2 Tampilan form untuk pakar menginputkan nama dan ciri jenis kepribadian

Langkah dalam pembuatan form sama seperti yang telah dibahas pada sub 5.3.1.1

Membuat *editor* nama jenis kepribadian

Kepribadian

Fungsinya adalah untuk memberikan nama dari jenis kepribadian sehingga bisa dibeda-bedakan jenisnya.

1. Klik *komponen pallete standard, edit* dan pindahkan ke form
2. Atur posisi sesuai dengan kebutuhan dan keinginan
3. Klik *object inspecor, text*, dan kosongkan nilainya

Ciri-Ciri

Membuat string grid

Fungsinya adalah untuk memberikan ciri-ciri dari setiap jenis kepribadian yang mungkin ada dari setiap orang.

1. Klik *komponen pallete additional, sting grid* dan pindahkan ke form
2. Klik *object inspecor, propertis, fixedcol* masukan nilai 0
3. Fixedrow=1
4. Klik *events onkeypress* isikan listing program berikut

```

procedure TF_input.StringGrid1KeyPress(Sender: TObject; var
Key: Char);
var a,b : integer;
c : string;
begin
if key = #13 then
begin
key := #0;
b := stringgrid1.row;
stringgrid1.RowCount := b + 2;
stringgrid1.Col := 0;
stringgrid1.Row := b + 1;
stringgrid1.SetFocus;
end;
end;

```

Membuat dbnavigator



Fungsinya adalah untuk melihat data sesudahnya dan sebelumnya selain itu juga untuk menyimpan dan menghapus data apabila seorang pakar selesai mengeditnya.

1. Klik *komponen pallete daa akses, db navigator*, dan pindahkan ke form
2. Klik *object insector, propertis, data source* dan isikan dengan kmpinen data source yang merujuk ke tabel yang dipakai (datasource1)
3. Pada event onklik masukan listing berikut

```

procedure TF_input.DBNavigator1Click(Sender: TObject;
  Button: TNavigateBtn);
var a : string;
    i : integer;
begin
  for i := 1 to (stringgrid1.RowCount - 1) do
  begin
    if stringgrid1.cells[0,i] <> '' then
      stringgrid1.cells[0,i] := '';
    end;
    stringgrid1.RowCount := 2;

    f_detail.Table1.close;
    f_detail.Table1.open;
    a := 'delete from ciri where diagnosa '+'
      'not in (select diagnosa from analisa)';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    i := 1;
    a := 'select ciri from ciri where diagnosa = '+#39+
      uppercase(table1.FieldName('diagnosa').asString)+#39;
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.open;
    stringgrid1.RowCount := 2;
    stringgrid1.Cells[0,i] := '';
    query1.FindFirst;
    while not query1.EOF do
      begin

```



```

        stringgrid1.Cells[0,i] :=
query1.fieldbyname('ciri').asString;
        stringgrid1.RowCount := stringgrid1.RowCount + 1;
        i := i + 1;
        query1.Next;
    end;
    stringgrid1.row := stringgrid1.RowCount - 1;
    stringgrid1.SetFocus;
    dbedit1.SetFocus;
end;

```

Detail Keterangan

Membuat tombol 'DETAIL KETERANGAN'

Fungsinya untuk masuk ke form selanjutnya yang berisi penginputan data tentang keterangan, penyebab dan solusi dari jenis kepribadian yang telah dimaksudkan dalam form ini. Cara pembuatan tombolnya dapat dilihat subbab 5.3.1.1 tetapi pada event onklik masukan listing berikut

```

procedure TF_input.Button2Click(Sender: TObject);
var i : integer;
begin
    f_detail.ShowModal;
    {for i := 1 to (stringgrid1.RowCount - 1) do
    begin
        if (stringgrid1.cells[0,i] <> '') then
            stringgrid1.cells[0,i] := '';
        end;
    }
    stringgrid1.RowCount := 2;
end;

```

Selain tombol diatas masih ada satu tombol lagi yaitu tombol 'KELUAR' yang fungsi dan cara pembuatannya dapat dilihat pada subbab 5.3.1.1.

Langkah mengaktifkan komponen query

1. klik dari komponen pallette data akses, query dan pindahkan ke form
2. klik object inspector, data base name dan isikan TIKTA (alias data base yang dipakai program)

Penggunaan komponen tabel

1. klik komponen pallette data akses , tabel, pindahkan ke form

2. klik object inspector, propertis, data base name dan isikan nama TIKA (alias data base yang dipakai)
3. klik object inspector, propertis, tabel name dan isikan dengan label yang akan dipakai (solusi db)

Langkah penggunaan komponen data source

1. klik komponen data akses, data source, dan pindahkan ke form
2. klik object inspector, propertis, data set dan isikan dengan nama komponen tabel yang kita pakai (table1)

5.3.1.3 Implementasi Form Untuk Pakar Menginputkan Detail Keterangan Jenis Kepribadian

Proses ini digunakan seorang pakar untuk pengisian keterangan dari kepribadian, penyebab dari kepribadian tersebut dan solusi dari jenis kepribadian yang dimaksudkan. Pada proses ini seorang pakar akan dituntut harus benar-benar teliti karena anjurannya akan dipakai sebagai pedoman dari pengambilan keputusan. Tampilan detail informasi tentang jenis kepribadian manusia dapat dilihat pada gambar 5.3.

Detail Informasi Problem

Nama

Keterangan

Penyebab

Solusi

Keluar

Gambar 5.3 Tampilan form untuk pakar menginputkan detail keterangan penyebab, dan solusi kepribadian

Membuat *db edit* Nama

Fungsinya adalah untuk menampilkan nama dari jenis kepribadian yang telah disimpan.

1. Klik *komponen pallete data control, db edit* dan pindahkan ke form
2. Atur posisi sesuai kebutuhan
3. Klik *object insector, propertis, data source* dan isikan dengan kompinen data source yang merujuk ke tabel yang dipakai (*datasource1*)
4. Klik *object insector, propertis, data field* dan isikan dengan kolom tabel yang akan dipakai(*nama*)

Langkah pembuatan *db memo1*

Keterangan

db memo2.

Penyebab

dan *db memo3* sama seperti dalam pembuatan *db edit*.

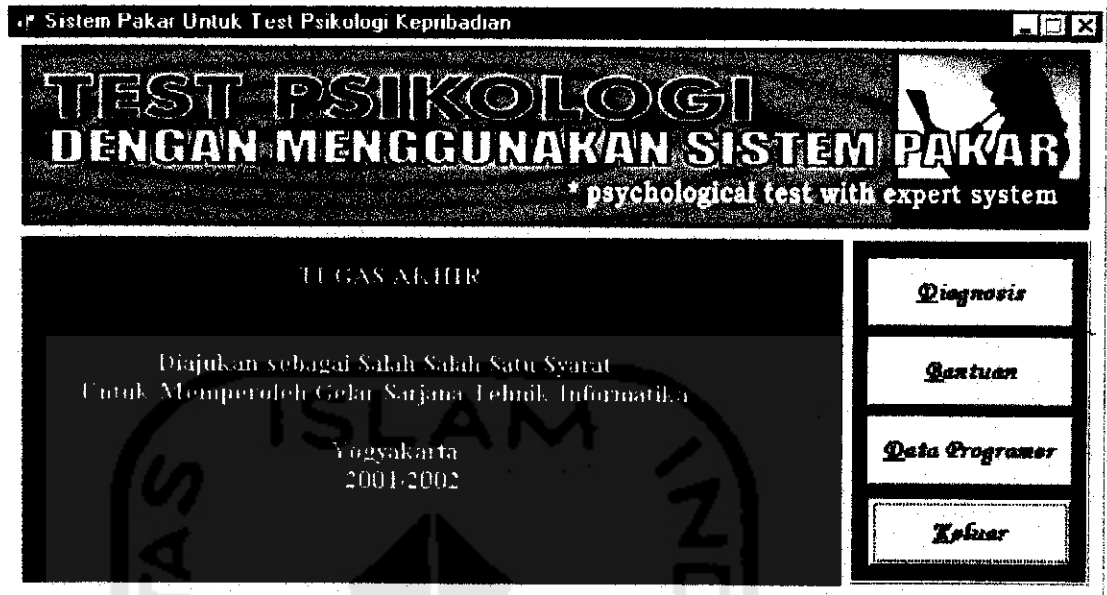
Solusi

Selain itu juga terdapat tombol **KELUAR** yang cara pembuatannya dapat dilihat pada subbab 5.3.1.1 yang sudah diuraikan secara jelas.

5.3.2 Implementasi Untuk Konsultasi

5.3.2.1 Implementasi Form Tampilan Awal untuk Konsultasi

Tampilan awal dari test psikologi dengan menggunakan sistem pakar terlihat pada gambar 5.4



Gambar 5.4 Tampilan form awal untuk *konsultasi*

Langkah dalam membuat form, nama form, tombol dan image dapat dilihat dalam

subbab 5.3.1.1 tetapi dalam pembuatan tombol 'DIAGNOSIS'



pada event on klik masukan listing berikut

```
procedure Tf_muka.Button4Click(Sender: TObject);
begin
  button4.Enabled := false;
  button1.Enabled := false;
  button2.Enabled := false;
  button3.Enabled := false;
  f_konsol.ShowModal;
end;
```

5.3.2.2 Implementasi Form untuk Konsultasi Menginputkan Ciri Awal yang

Diketahui

Proses identifikasi ciri awal digunakan untuk seorang pengguna yang mengetahui jenis kepribadiannya. Pada proses ini terdapat banyaknya masalah yang

diketahui oleh orang yang akan berkonsultasi dan ciri-ciri kepribadian yang akan ditanyakan. Tampilan dari identifikasi fakta awali terlihat pada gambar 5.5

Gambar 5.5 Tampilan form untuk konsultasi menginputkan ciri awal yang diketahui

Langkah dalam pembuatan form sama seperti pada subbab 5.3.1.1 sedangkan untuk pembuatan tombol **KELUAR**, query, string grid dapat dilihat dalam subbab 5.3.1.2.

Membuat tombol **BATAL**

Fungsinya adalah untuk membatalkan proses konsultasi yang mungkin disebabkan karena mengalami keraguan dalam penentuan identifikasi awalnya

1. Dari *event on klik* isikan perintah berikut

```

procedure TF_konsol.Button2Click(Sender: TObject);
var i : integer;
a : string;
begin
  for i := 1 to (stringgrid1.RowCount - 1) do
  begin
    if stringgrid1.cells[0,i] <> ''then
      stringgrid1.cells[0,i] := '';
    end;

    a := 'delete from proc_analisis1';
  
```

```

query1.close;
query1.sql.Clear;
query1.sql.Add(a);
query1.ExecSQL;

a := 'delete from proc_analisis2';
query1.close;
query1.sql.Clear;
query1.sql.Add(a);
query1.ExecSQL;

a := 'delete from ciri_tmp';
query1.close;
query1.sql.Clear;
query1.sql.Add(a);
query1.ExecSQL;

a := 'delete from ciri_tmp1';
query1.close;
query1.sql.Clear;
query1.sql.Add(a);
query1.ExecSQL;

stringgrid1.RowCount := 2;
end;

```

Diagnosa

Membuat tombol DIAGNOSA

Tombol ini berbeda fungsinya dengan tombol diagnosa yang sudah dibahas di atas tetapi tombol ini mempunyai fungsi untuk mengecek fakta yang telah dituliskan ada dalam data base atau tidak pada *eventinklikbutton cek* isikan perintah berikut

```

procedure TF_konsol.Button1Click(Sender: TObject);
var a : string;
i : integer;
begin {Mengecek keterkaitan fakta dengan basis pengetahuan}
  for i := 1 to (stringgrid1.RowCount - 1) do
    begin
      if stringgrid1.cells[0,i] <> '' then
        begin
          a := 'select * from ciri where ciri =
'+#39+uppercase(stringgrid1.Cells[0,i])+#39;
          query1.close;
          query1.sql.Clear;
          query1.sql.Add(a);
          query1.open;

          if query1.RecordCount > 0 then
            begin

```

```

        a := 'insert into ciri_tmp values
('+#39+uppercase(stringgrid1.Cells[0,i])+#39+')';
        query1.close;
        query1.sql.Clear;
        query1.sql.Add(a);
        query1.ExecSQL;
        end;
        stringgrid1.cells[0,i] := '';
        end;
    end;

a := 'select * from ciri_tmp';
query1.close;
query1.sql.Clear;
query1.sql.Add(a);
query1.open;

if query1.RecordCount < 1 then
begin
    a := 'insert into proc_analisis2 select distinct(diagnosa)
from ciri';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;
end
else
begin
    a := 'insert into proc_analisis2 select distinct(diagnosa)
from ciri where ciri in '+
'(select ciri from ciri_tmp)';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;
end;

a := 'select * from ciri_tmp';
query2.close;
query2.sql.Clear;
query2.sql.Add(a);
query2.open;

if query2.RecordCount > 0 then
begin
    query2.FindFirst;
    while not query2.EOF do
    begin
        a := 'delete from proc_analisis1';
        query1.close;
        query1.sql.Clear;
        query1.sql.Add(a);
        query1.ExecSQL;

        a := 'insert into proc_analisis1 select distinct(diagnosa)
from ciri where ciri = '+'

```



```

        #39+query2.fieldbyname('ciri').asstring+#39+
        'and diagnosa in (select diagnosa from
proc_analisis2)';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'delete from proc_analisis2';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'insert into proc_analisis2 select distinct(diagnosa)
from ciri where ciri = '+
        #39+query2.fieldbyname('ciri').asstring+#39+
        'and diagnosa in (select diagnosa from
proc_analisis1)';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    query2.Next;
end;
else
begin
    a := 'delete from proc_analisis1';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'insert into proc_analisis1 select distinct(diagnosa)
from ciri where '+
        'diagnosa in (select diagnosa from proc_analisis2)';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'delete from proc_analisis2';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'insert into proc_analisis2 select distinct(diagnosa)
from ciri where '+
        'diagnosa in (select diagnosa from proc_analisis1)';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

```

```

end;

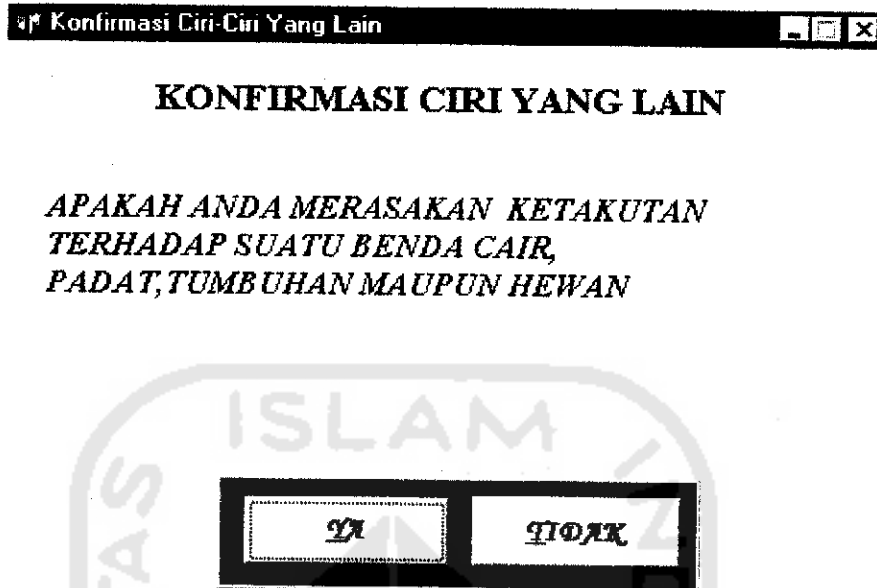
a := 'select * from proc_analisis1';
query2.close;
query2.sql.Clear;
query2.sql.Add(a);
query2.open;
if query2.RecordCount > 0 then
begin
if query2.RecordCount > 1 then
begin
f_tanya.ShowModal;
end
else
begin
f_tanya.kosong := false;
f_info.ShowModal;
end;
end
else
begin
f_tanya.kosong := true;
f_info.ShowModal;
end;
stringgrid1.RowCount := 2;
end;

```

5.3.2.3 Implementasi Form untuk Konsultasi Mengkonfirmasi dengan Ciri Awal yang Bersangkutan

Proses klarifikasi ciri yang bersangkutan digunakan untuk mengklarifikasi ciri-ciri kepribadian yang sedang dialami dan tidak diketahui oleh orang yang akan berkonsultasi. Pada proses terdapat beberapa pertanyaan yang harus dijawab. Pertanyaan yang akan ditanyakan oleh sistem berkaitan dengan kepribadian manusia. Pertanyaan ini dimaksudkan untuk menyakinkan pengguna tentang ciri-ciri yang lainnya yang kemudian akan di diagnosa oleh sistem. Tampilan klarifikasi ciri yang berkaitan terlihat pada gambar 5.6

Langkah untuk pembuatan query dapat dilihat dalam subbab 5.3.1.2.



Gambar 5.6 Tampilan form untuk konsultasi mengkonfirmasi dengan ciri awal yang bersangkutan

Membuat komponen memo untuk menampilkan pertanyaan

*APAKAH ANDA MERASAKAN KETAKUTAN
TERHADAP SUATU BENDA CAIR,
PADAT, TUMBUHAN MAUPUN HEWAN*

Fungsinya adalah untuk menampilkan pertanyaan fakta yang berkaitan dengan fakta awal yang berkaitan.

1. Klik *komponen pallet standard, memo* dan pindahkan ke form
2. Klik *object inspector, properti, lines, string* dan bersihkan nilainya
3. Pada *properti name* berikan nama sesuai yang diinginkan

Langkah dalam pembuatan form, nama form dapat dilihat dalam subbab 5.3.1.1

hanya pada *event on activate* isikan listing program berikut:

```

procedure TF_konsol.FormActivate(Sender: TObject);
var a : string;
begin
  a := 'delete from penyakit_tmp';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'delete from penyakit_tmp1';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'delete from ciri_tmp';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'delete from ciri_tmp1';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  edit1.Clear;
  stringgrid1.Cells[0,0] := 'Ciri-Ciri';
  button1.Enabled := false;
  stringgrid1.Enabled := true;
  edit1.SetFocus;
end;

```


 YA

TIDAK

Membuat komponen *radio group*

Fungsinya adalah untuk menjawab pertanyaan yang diberikan, jawaban hanya tinggal mengklik salah satu tombol, tombol ya atau tidak.

1. Klik *komponen palette standard, radiogroup*, dan pindahkan e form
2. Klik *object inspector, propertis, columns* isikan angka 2
3. Pada *propertis items* isikan nilai ya, tidak
4. Pada *event onklik* di tombol YA isikan listing program berikut:

```

procedure Tf_tanya.Button1Click(Sender: TObject);
var a : string;

```

```

begin
  a := 'delete from proc_analisis2';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'insert into ciri_tmp values
('+#39+query2.fieldbyname('ciri').asstring+#39+')';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'insert into proc_analisis2 select distinct(diagnosa)
from ciri where ciri = '+
      '#39+query2.fieldbyname('ciri').asstring+#39+
      'and diagnosa in (select diagnosa from
proc_analisis1)';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'delete from proc_analisis1';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'insert into proc_analisis1 select * from
proc_analisis2';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'delete from ciri_tmp';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'insert into ciri_tmp select distinct(ciri) from ciri
where diagnosa'+
      ' in (select diagnosa from proc_analisis1) and ciri
'+
      ' not in (select ciri from ciri_tmp)';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'select * from ciri_tmp';
  query2.close;
  query2.sql.Clear;

```

```

query2.sql.Add(a);
query2.open;
query2.FindFirst;
if query2.fieldbyname('ciri').asstring <> '' then
begin
    memol.Lines.Text := 'APAKAH ANDA MERASAKAN ' +
query2.fieldbyname('ciri').asstring;
end
else
begin
    f_info.ShowModal;
end;
end;

```

5. Pada *event onklik* di tombol **TIDAK** isikan listing program berikut

```

procedure Tf_tanya.Button2Click(Sender: TObject);
var a : string;
begin
    a := 'delete from proc_analisis2';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'delete from proc_analisis1 where diagnosa in '+
        '(select distinct(diagnosa) from ciri where ciri = '+
        #39+query2.fieldbyname('ciri').asstring+#39+
        'and diagnosa in (select diagnosa from
proc_analisis1))';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'insert into proc_analisis2 select * from
proc_analisis1';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'delete from ciri_tmp';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);
    query1.ExecSQL;

    a := 'insert into ciri_tmp select distinct(ciri) from ciri
where diagnosa'+
        ' in (select diagnosa from proc_analisis1) and ciri '+
        ' not in (select ciri from ciri_tmp1)';
    query1.close;
    query1.sql.Clear;
    query1.sql.Add(a);

```

```

query1.ExecSQL;

a := 'select * from ciri_tmp';
query2.close;
query2.sql.Clear;
query2.sql.Add(a);
query2.open;
query2.FindFirst;
if query2.fieldbyname('ciri').asString <> '' then
begin
    mem01.Lines.Text := 'APAKAH ANDA MERASAKAN ' +
query2.fieldbyname('ciri').asString;
end
else
begin
    f_info.ShowModal;
end;
end;

{begin
    query2.Ncxt;
    if query2.fieldbyname('ciri').asString <> '' then
    begin
        if not query2.EOF then
        begin
            mem01.Lines.Text := 'APAKAH ANDA MERASAKAN ' +
query2.fieldbyname('ciri').asString;
        end
        else
        begin
            kosong := true;
            f_info.ShowModal;
        end;
    end;
end;}
end;}

```

5.3.3 Implementasi Form Tampilan Hasil Keluaran Sistem

Proses hasil keluaran sistem digunakan untuk menampilkan hasil dari diagnosa sistem. Proses ini berisi tentang nama jenis kepribadian yang diderita oleh manusia, keterangan dari jenis kepribadian tersebut, penyebab dari jenis kepribadian tersebut dan solusi dari jenis kepribadian tersebut. Tampilan hasil keluaran system terlihat pada gambar 5.7.

Informasi

Kepribadian : PHOBIA

Keterangan

adalah ketakutan atau kecemasan yang abnormal. tidak rasional dan tak bisa dikontrol terhadap sesuatu situasi atau obyek tertentu merupakan ketakutan khas

Penyebab

- pernah mengalami ketakutan yang hebat.
- pengalaman asli ini dibarengi oleh rasa malu dan rasa bersalah kemudian semuanya ditekan agar dapat

Solusi

Hindari rasa bersalah yang terlalu berlebihan dan segera hilangkan perasaan bersalah apabila kita telah melakukan suatu kesalahan walaupun berakibat fatal.

Tambah Fakta **Cetak** **Keluar**

Gambar 5.7 Tampilan form hasil keluaran sistem

Pembuatan form dan nama form dapat dilihat pada subbab 5.3.1.1 sedangkan untuk pembuatan dbmemo, buttom, query, data source, dan label dapat dilihat pada subbab 5.3.1.2.

Pada pembuatan form *event onclose* isikan listing program berikut

```

procedure Tf_info.FormClose(Sender: TObject; var Action:
TCloseAction);
var a : string;
begin
  a := 'delete from proc_analisis1';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.ExecSQL;

  a := 'delete from proc_analisis2';
  query1.close;
  query1.sql.Clear;

```



```

query1.sql.Add(a);
query1.ExecSQL;

a := 'delete from ciri_tmp';
query1.close;
query1.sql.Clear;
query1.sql.Add(a);
query1.ExecSQL;

a := 'delete from ciri_tmp1';
query1.close;
query1.sql.Clear;
query1.sql.Add(a);
query1.ExecSQL;

f_konsol.stringgrid1.RowCount := 2;
f_tanya.Close;
end;

```

Pada pembuatan form *event onshow* isikan listing program berikut

```

procedure Tf_info.FormShow(Sender: TObject);
var a : string;
begin
  a := 'select * from analisa where diagnosa = (select diagnosa
from proc_analisis1)';
  query1.close;
  query1.sql.Clear;
  query1.sql.Add(a);
  query1.open;

  if query1.RecordCount > 0 then
    f_tanya.kosong := false
  else
    f_tanya.kosong := true;

  if f_tanya.kosong = false then
    begin
      dbmemo1.Visible := true;
      dbmemo2.Visible := true;
      dbmemo3.Visible := true;
      label3.Visible := false;
      label1.Visible := true;
      label4.Visible := true;
      label5.Visible := true;
      label2.Visible := true;
      dbtext1.Visible := true;
    end
  else
    begin
      dbmemo1.Visible := false;
      dbmemo2.Visible := false;
      dbmemo3.Visible := false;
      label3.Visible := true;
      label1.Visible := false;
    end
  end;
end;

```

```
label4.Visible := false;  
label5.Visible := false;  
label2.Visible := false;  
dbtext1.Visible := false;  
end;  
end;
```

Pembuatan komponen dbtext **Kepribadian : PHOBLA**

1. Klik *komponen pallete data control, dbtext* dan pindahkan ke form
2. Klik *object inspector, propertis, data source* dan isikan dengan *fata source* yang merujuk ke tabel data base yang dipakai

